Variables, datatypes and operators

Table of Contents

Comments

A comment is text that is ignored by the PHP engine. Comments help humans understand the code. When learning a new language adding comments can seam pointless. However, when developing an application comments are vital. An application is reviewed, enhanced, debugged and comments help the programmers understand the code.

Comments can be added by:

- //this is a comment and ends at the end of the line
- #this is also a comment and ends at the end of the line
- /*this is also a comment and can span several lines ending with */

Not commenting your code is extremely bad practice, get into the habit now!

Variables

Variables are fundamental to programming; they let you assign a label to an area of memory in which you can store just about any type of information. A variable is a container that contains a value, and the value can be changed.

Using a variable in PHP is made up of 2 parts: the variable name and the value to be stored in the variable.

- Variable names start with a $ sign.
- The first character after the $ sign must be a letter or an underscore,
- The remaining characters in the name can be letters, numbers, or underscores.
- Variable names are case sensitive

Creating variables

Creating a variable (called declaring a variable) is done automatically when PHP first sees a variable name in the script. When you declare a variable in PHP, it is often good practice to assign a value to it at the same time. This is known as initializing a variable. If no value is assigned it's given the value of null.

Example:

```
$age = 3;
```

This creates a variable called $age and assigns the value 3 to it by using the "=" sign.

Data types

All data stored in PHP is one of 8 data types. A variable's data type determines what operations can be carried out on the variable's data, as well as the amount of memory needed to hold the data.

Data type	Description
Integer	A whole number
Float	A floating-point number
String	A series of characters
Boolean	True or False
Array	A collection of values
Object	A type that contains properties and methods
Resource	Reference to an external resource such as a file or database
Null	Does not contain a value

PHP is known as a loosely typed language. This means you don't need to specify the data type of the variable when you declare it. PHP determines the data type based on the value assigned to the variable. PHP also converts a variable's data type automatically, depending on the context in which the variable is used. For example, you can initialize a variable with an integer value; add a float value to it (thereby turning it into a float); then join it onto a string value (turning it into a string).

Sometimes it is necessary to check or change the type manually.

Activity 1. Variables

Create a PHP file that creates 2 variable one to hold the course name and one to hold the number of people in your class. Display both of these variables.

Changing the type of a variable

You can use settype() to set the type of a variable:

```
settype($students, "string");
```

Changing type by casting

You can also cause a variable's value to be treated as a specific type using a technique known as type casting. This involves placing the name of the desired data type in parentheses before the variable's name. Note that the variable itself remains unaffected; this is in contrast to settype() , which changes the variable's type.

```
(string)$students
```

You can use casting for:

- int or integer
- float
- string
- bool or boolean
- array
- object

Operators

An operator is something that takes one or more values and creates a new value.

Assignment operator

The assignment operator (=) is used to assign a value to a variable. The equals sign (=) can be combined with other operators to give you a combined assignment operator that makes it easier to write certain expressions. The combined assignment operators (such as +=, –=, and so on) simply give you a shorthand method for performing typical arithmetic operations, so that you don't have to write out the variable name multiple times. For example, you can write:

```
$first_number += $second_number;
```

rather than:

```
$first_number = $first_number + $second_number;
```

Arithmetic operators

Symbol	Operator	Example
+	Addition	1 + 2 = 3
-	Subtraction	2 - 1 = 1
*	Multiplication	2 * 4 = 8
/	Division	8 / 2 = 4
%	Modulus (remainder after a division)	6 % 3 = 0 (remainder of 6/3)

Incrementing/decrementing operators

In programming it's common to add 1 or subtract 1 from a value. In PHP you can use:

- $x++; Returns $x then adds 1 to it
- ++$x; Adds 1 to $x then returns $x
- $x--; Returns $x then subtracts 1 from it
- --$x; Subtract 1 from $x then returns $x

Example:

```
$x = 5;
echo ++$x;   // Displays "6" (and $x now contains 6)
$x = 5;
echo $x++;   // Displays "5" (and $x now contains 6)
```

Concatenating strings

Combining a string with another string is known as string concatenation. The dot (.) operator can be used for string concatenation.

Example:

```
$a = "Hello ";
$b = $a . "World!"; // now $b contains "Hello World!"
```

You can also use it with the = sign

```
$a = "Hello ";
$a .= "World!";     // now $a contains "Hello World!"
```

Activity 2. Concatenating strings

Create a file with 2 variables one to hold your first name and the second to hold your surname. Display both variables separated by a space.

Constants

Constants are like variables in that they can hold a value, but unlike variables the value cannot change. Constants can be defined only once in a PHP program. Constants do not start with $ sign and it is good practice to use all capital letters for the name of a constant.

Constants may only contain scalar values such as boolean, integer, float, and string (not values such as arrays and objects), and can be used from anywhere in your PHP program. To define a constant, use the define() function.

Example:

```
define( “CAMPUSNAME”, “Hornsby” );
echo CAMPUSNAME;  //will display “Hornsby”
```

Exercises

1. Write a PHP script that creates a variable and assigns an integer value to it, then adds 1 to the variable’s value three times, using a different operator each time. Display the final result to the user.